

EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	TTTTTTTTTTTTTTTT
EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	TTTTTTTTTTTTTTTT
EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	TTTTTTTTTTTTTTTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEEEEEEEEEEEEE	DDD	TTT
EEEEEEEEEEEEEE	DDD	TTT
EEEEEEEEEEEEEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	TTT
EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	TTT
EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	TTT



```
0001 0 ZTITLE 'EDT$SCRRLIN - refresh a screen line'
0002 0 MODULE EDT$SCRRLIN ( ! Refresh a screen line
0003 0 IDENT = 'V04-000' ! File: SCRRLIN.BLI Edit: REM1034
0004 0 ) =
0005 1 BEGIN
0006 1
0007 1 *****
0008 1 *
0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0011 1 * ALL RIGHTS RESERVED.
0012 1 *
0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0018 1 * TRANSFERRED.
0019 1 *
0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0022 1 * CORPORATION.
0023 1 *
0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0026 1 *
0027 1 *****
0028 1
0029 1
0030 1
0031 1 ++
0032 1 FACILITY: EDT -- The DEC Standard Editor
0033 1
0034 1 ABSTRACT:
0035 1
0036 1 This module refreshes a single line on the screen.
0037 1
0038 1 ENVIRONMENT: Runs at any access mode - AST reentrant
0039 1
0040 1 AUTHOR: Bob Kushlis, CREATION DATE: September 8, 1979
0041 1
0042 1 MODIFIED BY:
0043 1
0044 1 1-001 - Original. DJS 12-Feb-1981. This module was created by
0045 1 extracting the routine EDT$SC RFRELN from module SCREEN.
0046 1 1-002 - Regularize headers. JBS 13-Mar-1981
0047 1 1-003 - Change [EOB] to user defined string STS 06-Oct-1981
0048 1 1-004 - Do an absolute cursor position before writing the blob at
0049 1 end of line, to avoid running off the edge of the screen.
0050 1 Also, show the blob only if the text exceeds the screen
0051 1 width. JBS 02-Apr-1982
0052 1 1-005 - Show characters all the way to end edge of the screen. JBS 06-Apr-1982
0053 1 1-006 - Worry about wide characters at the edge of the screen. JBS 15-Apr-1982
0054 1 1-007 - Continue work on edit 1-006. JBS 16-Apr-1982
0055 1 1-008 - Always show [EOB] (or whatever text it has been set to) in non-reverse
0056 1 video. JBS 16-Apr-1982
0057 1 1-009 - Make the edge of the screen logic work on a VT100, which clears its
```



```
58 0058 1 1 wrap flag only when a character is printed. JBS 19-Apr-1982
59 0059 1 1-010 - Don't erase the message lines if an error occurs during select.
60 0060 1 1 SMB 01-Jul-1982
61 0061 1 1-011 - Fix bug introduced by edit 1-010. SMB 20-Jul-1982
62 0062 1 1-012 - Add check for message flag to erasure of screen. SMB 23-Jul-1982
63 0063 1 1-013 - Change the flag checked in edit 1-012. SMB 28-Jul-1982
64 0064 1 1-014 - Go back to edit 1-012. SMB 17-Aug-1982
65 0065 1 1-015 - Modify for the new screen updater. SMB 24-Sep-1982
66 0066 1 1-016 - Simplify for the new screen update logic. This version always repaints
67 0067 1 1 any changed line. JBS 30-Sep-1982
68 0068 1 1-017 - Remove unused external declaration of EDT$FMT_LIT. JBS 05-Oct-1982
69 0069 1 1-018 - Fix painting of select range. JBS 08-Oct-1982
70 0070 1 1-019 - Put call to fsetcol in line. STS 11-Oct-1982
71 0071 1 1-020 - Start work on NOTRUNCATE mode. JBS 11-Oct-1982
72 0072 1 1-021 - Debug NOTRUNCATE mode. JBS 12-Oct-1982
73 0073 1 1-022 - Fix the call to EDT$FMT_CHWID. JBS 13-Oct-1982
74 0074 1 1-023 - Add the second argument. JBS 23-Oct-1982
75 0075 1 1-024 - Use SCR_EDIT_MINPOS. JBS 28-Oct-1982
76 0076 1 1-025 - Be sure to print at least one character before the last character
77 0077 1 1 of a line, so we won't be hit by the VT100's autowrap. JBS 10-Nov-1982
78 0078 1 1-026 - Set the final MINPOS to CHR TO, so CHMEINPUT's text won't have to be rewritten. JBS 02-Dec-1982
79 0079 1 1-027 - Change the handling of EDT$G SHF. JBS 14-Dec-1982
80 0080 1 1-028 - Maintain and use SCR_EDIT_MAXPOS. JBS 27-Dec-1982
81 0081 1 1-029 - Don't erase to end of line if we do not repaint the whole line. JBS 27-Dec-1982
82 0082 1 1-030 - Put the most common cases of character formatting in-line, to improve speed. JBS 04-Jan-1983
83 0083 1 1-031 - Be sure the blob is painted with correct video attributes. JBS 21-Mar-1983
84 0084 1 1-032 - Make sure we are in replace mode. JBS 01-Apr-1983
85 0085 1 1-033 - Adjust the width of a tab if it is at the front of a continued line. JBS 03-May-1983
86 0086 1 1-034 - Fix bug where if the EOB marker displays in the last column of the
87 0087 1 1 screen, it was deleted when we attempted to delete to end of line.
88 0088 1 1 The bug happened only if advancing to that line without clearing the
89 0089 1 1 screen first. REM 12-Dec-1983
90 0090 1 1 --
91 0091 1 1
```

EDTSSCRRLIN  
V04-000

EDTSSCRRLIN - refresh a screen line  
Declarations

F 10  
16-Sep-1984 01:42:29  
14-Sep-1984 12:24:38

VAX-11 BLISS-32 V4.0-742  
DISK\$VMSMASTER:[EDT.SRC]SCRRLIN.BLI;1  
Page 3  
(2)

```

93 0092 1 $SBTTL 'Declarations'
94 0093 1
95 0094 1 TABLE OF CONTENTS:
96 0095 1
97 0096 1
98 0097 1 REQUIRE 'EDTsrc:TRAROUNAM';
99 0536 1
100 0537 1 FORWARD ROUTINE
101 0538 1 EDTSSC_RFRELN : NOVALUE;
102 0539 1
103 0540 1
104 0541 1 INCLUDE FILES:
105 0542 1
106 0543 1
107 0544 1 REQUIRE 'EDTsrc:EDTREQ';
108 0679 1
109 0680 1
110 0681 1 MACROS:
111 0682 1
112 0683 1 NONE
113 0684 1
114 0685 1 EQUATED SYMBOLS:
115 0686 1
116 0687 1 NONE
117 0688 1
118 0689 1 OWN STORAGE:
119 0690 1
120 0691 1 NONE
121 0692 1
122 0693 1 EXTERNAL REFERENCES:
123 0694 1
124 0695 1 In the routine
```

```
126 0696 1 XSBTTL 'EDTSSC_RFRELN - refresh a line on the screen'
127 0697 1
128 0698 1 GLOBAL ROUTINE EDTSSC_RFRELN (
129 0699 1     SCRPTR,
130 0700 1     ERASED
131 0701 1     ) : NOVALUE =
132 0702 1
133 0703 1 ++
134 0704 1 FUNCTIONAL DESCRIPTION:
135 0705 1
136 0706 1     This routine refreshes a single line on the screen. It expects EDTSSG_CS_LNO
137 0707 1     to be the screen line number to be refreshed. This routine operates only on
138 0708 1     the specified line; it does not clear the screen after an [EOB], for example.
139 0709 1
140 0710 1 FORMAL PARAMETERS:
141 0711 1
142 0712 1     SCRPTR          Pointer to the screen block for the line being refreshed
143 0713 1
144 0714 1     ERASED          1 = the line has already been erased
145 0715 1
146 0716 1 IMPLICIT INPUTS:
147 0717 1
148 0718 1     EDTSSG_CS_LNO
149 0719 1     EDTSSA_SEC_BUF
150 0720 1     EDTSSG_SHF
151 0721 1     EDTSSG_TI_WID
152 0722 1     EDTSSA_WK_LN
153 0723 1     EDTSSG_FMT_LNPOS
154 0724 1     EDTSSA_CUR_TBCB
155 0725 1     EDTSSA_EOB_SCRPTR
156 0726 1     EDTSSA_FMT_CUR
157 0727 1     EDTSSG_PRV_COL
158 0728 1     EDTSST_FMT_BUF
159 0729 1     EDTSSG_INSERT_MODE
160 0730 1
161 0731 1 IMPLICIT OUTPUTS:
162 0732 1
163 0733 1     EDTSSA_FMT_CUR
164 0734 1     EDTSSG_PRV_COL
165 0735 1
166 0736 1 ROUTINE VALUE:
167 0737 1
168 0738 1     NONE
169 0739 1
170 0740 1 SIDE EFFECTS:
171 0741 1
172 0742 1     Writes on the screen.
173 0743 1
174 0744 1 --
175 0745 1
176 0746 2 BEGIN
177 0747 2
178 0748 2 EXTERNAL ROUTINE
179 0749 2     EDTSSFMT_CH : NOVALUE,
180 0750 2     EDTSSFMT_CHWID,
181 0751 2     EDTSSC_SHWBLOB : NOVALUE,
182 0752 2     EDTSSC_REVIDCHK : NOVALUE,
183 0753 2
184 0754 2     ! Output a character
185 0755 2     ! Compute the width of a character
186 0756 2     ! Output a blob
187 0757 2     ! Check for reverse video based on select region
```



```
183 0753 EDTSSSC_NONREVID : NOVALUE,      ! Go to normal video
184 0754 EDTSSSC_POSCSIF : NOVALUE,      ! Position the cursor
185 0755 EDTSSSC_ERATOEOI : NOVALUE,      ! Erase to end of line
186 0756 EDTSSSC_ERAALL : NOVALUE,        ! Erase to end of screen
187 0757 EDTSSFMT_TEXT : NOVALUE,         ! Print [EOB]
188 0758 EDTSSOUT_FMTBUF : NOVALUE,        ! Output the format buffer
189 0759 EDTSSSC_REP_MODE : NOVALUE;      ! Put the screen in replace mode
190 0760
191 0761
192 0762 EXTERNAL
193 0763 EDTSSA_EOB_SCRPTR : REF SCREEN_LINE, ! Special line block for [EOB]
194 0764 EDTSSG_CS_LNO, ! current screen line
195 0765 EDTSSA_SEC_BUF, ! select buffer.
196 0766 EDTSSG_SHF, ! The number of columns shifted.
197 0767 EDTSSG_TI_WID, ! Width of terminal line.
198 0768 EDTSSA_WK_LN : REF LIN_BLOCK, ! Current line pointer.
199 0769 EDTSSG_FMT_LNPOS, ! Current column number
200 0770 EDTSSA_CUR_BUF : REF TBCB_BLOCK, ! Pointer to current text control block
201 0771 EDTSSA_FMT_CUR, ! Pointer to next char in output buffer
202 0772 EDTSSG_FMT_BUF : BLOCK [CH$ALLOCATION (EDTSSK_FMT_BUFLN)], ! Output buffer
203 0773 EDTSSG_PRV_COL, ! The cursor column number
204 0774 EDTSSG_INSERT_MODE; ! 1 = screen is in insert mode
205 0775
206 0776 MAP
207 0777 SCRPTR : REF SCREEN_LINE;
208 0778
209 0779 LOCAL
210 0780 TXTPTR,
211 0781 ORIG_TXTPTR,
212 0782 LEN,
213 0783 CHAR,
214 0784 CHAR_WIDTH,
215 0785 LEFT,
216 0786 FIRST_CHAR,
217 0787 WIDTH,
218 0788 SIMPLE_CHAR,
219 0789 MAXPOS;
220 0790
221 0791 !+ Make sure we are in replace mode.
222 0792 !-
223 0793
224 0794 IF (.EDTSSG_INSERT_MODE NEQ 0) THEN EDTSSSC_REP_MODE ();
225 0795
226 0796 !+ Check for EOB.
227 0797 !-
228 0798
229 0799
230 0800 IF (.SCRPTR EQLA .EDTSSA_EOB_SCRPTR)
231 0801 THEN
232 0802 BEGIN
233 0803 EDTSSSC_POSCSIF (.EDTSSG_CS_LNO, 0);
234 0804 EDTSSSC_NONREVID ();
235 0805 EDTSSFMT_TEXT (0);
236 0806
237 0807
238 0808 IF (( NOT .ERASED) AND (.SCRPTR [SCR_EDIT_MAXPOS] EQL 255))
239 0809 THEN ! If not erased and not at end of the line,
```

```
240 0810 4 BEGIN
241 0811 4 !! EDT$SC_POSCSIF (.EDT$G_CS_LNO, MAX (0, .EDT$G_FMT_LNPOS - .EDT$G_SHF));
242 0812 4 !! EDT$SC_ERATOEOI () ! erase any extra characters that may
243 0813 4 !! ! have been left on the screen's line.
244 0814 4 END;
245 0815 4
246 0816 4 +
247 0817 4 ! Mark the line as finished with its edit.
248 0818 4 -
249 0819 4 SCRPT [SCR_EDIT_MINPOS] = 255;
250 0820 4 SCRPT [SCR_EDIT_MAXPOS] = 0;
251 0821 4 SCRPT [SCR_EDIT_FLAGS] = .SCRPT [SCR_EDIT_FLAGS] AND ( NOT (SCR_EDIT_MODIFY OR SCR_EDIT_INSLN));
252 0822 4 RETURN;
253 0823 4 END;
254 0824 4
255 0825 4 +
256 0826 4 ! Not EOB. Position to the first character to be updated in the line,
257 0827 4 ! keeping track of the screen column which it will occupy.
258 0828 4 -
259 0829 4 WIDTH = .EDT$G_T1_WID + .EDT$G_SHF;
260 0830 4 LEFT = .SCRPT [SCR_CHR_FROM];
261 0831 4 LEN = MIN (.SCRPT [SCR_CHR_TO] + 1, .EDT$A_WK_LN [LIN_LENGTH]) - .LEFT;
262 0832 4 TXTPTR = CH$PLUS (EDT$A_WK_LN [LIN_TEXT], .LEFT);
263 0833 4 ORIG_TXTPTR = .TXTPTR;
264 0834 4 EDT$G_FMT_LNPOS = 0;
265 0835 4 CHAR = CH$RCHAR_A (TXTPTR);
266 0836 4
267 0837 4 IF ((.CHAR GEQ 'X'20') AND (.CHAR LEQ 'X'7E'))
268 0838 4 THEN
269 0839 4 BEGIN
270 0840 4 CHAR_WIDTH = 1;
271 0841 4 SIMPLE_CHAR = 1;
272 0842 4 END
273 0843 4 ELSE
274 0844 4 BEGIN
275 0845 4 CHAR_WIDTH = EDT$FMT_CHWID (.CHAR, .EDT$G_FMT_LNPOS);
276 0846 4 SIMPLE_CHAR = 0;
277 0847 4 END;
278 0848 4
279 0849 4 +
280 0850 4 ! Skip over unmodified characters on this line.
281 0851 4 -
282 0852 4
283 0853 4 WHILE (((.TXTPTR - .ORIG_TXTPTR) LEQ .SCRPT [SCR_EDIT_MINPOS]) AND !
284 0854 4 (.LEN GTR 0) AND
285 0855 4 (.EDT$G_FMT_LNPOS LSS (.WIDTH - .CHAR_WIDTH - 1))) DO
286 0856 4 BEGIN
287 0857 4 +
288 0858 4 ! Account for the blob at the front of continued lines.
289 0859 4 -
290 0860 4
291 0861 4 IF ((.EDT$G_FMT_LNPOS EQL 0) AND (.SCRPT [SCR_LINE_IDX] NEQ 0))
292 0862 4 THEN
293 0863 4 BEGIN
294 0864 4 +
295 0865 4 ! Adjust for the blob at the front of a continued line. This code requires
296 0866 4 ! that the shift amount always be a multiple of 8, so that shifting doesn't
```



```
297 0867 4 ! change tab stops.
298 0868 4 !-
299 0869 4     EDT$$G_FMT_LNPOS = .EDT$$G_SHF + 2;
300 0870 4
301 0871 3     IF (.CHAR EQL ASC_K_TAB)
302 0872 4     THEN
303 0873 3         BEGIN
304 0874 3         CHAR_WIDTH = .CHAR_WIDTH - 2;
305 0875 3         ASSERT (.CHAR_WIDTH EQL 6);
306 0876 4         END;
307 0877 4
308 0878 4     END;
309 0879 4
310 0880 4     EDT$$G_FMT_LNPOS = .EDT$$G_FMT_LNPOS + .CHAR_WIDTH;
311 0881 4     LEN = .LEN - 1;
312 0882 4     CHAR = CH$RCHAR_A (TXTPTR);
313 0883 4
314 0884 4     IF ((.CHAR GEQ ZX'20') AND (.CHAR LEQ ZX'7E'))
315 0885 3     THEN
316 0886 4         BEGIN
317 0887 4         CHAR_WIDTH = 1;
318 0888 4         SIMPL_CHAR = 1;
319 0889 4         END
320 0890 3     ELSE
321 0891 4         BEGIN
322 0892 4         CHAR_WIDTH = EDT$$FMT_CHWID (.CHAR, .EDT$$G_FMT_LNPOS);
323 0893 4         SIMPL_CHAR = 0;
324 0894 4         END;
325 0895 4
326 0896 4     END;
327 0897 4
328 0898 4 + Put the characters into the format buffer.
329 0899 4 -
330 0900 4     FIRST_CHAR = 1;
331 0901 4
332 0902 4 + If this is a continued line, indicate this at the front of the line.
333 0903 4 -
334 0904 4
335 0905 4     IF ((.SCRPTR [SCR_LINE_IDX] NEQ 0) AND (.EDT$$G_FMT_LNPOS EQL 0))
336 0906 4     THEN
337 0907 4         BEGIN
338 0908 4         EDT$$G_FMT_LNPOS = .EDT$$G_SHF;
339 0909 4         EDT$$$C_POSCSIF (.EDT$$G_CS_LNO, .EDT$$G_FMT_LNPOS - .EDT$$G_SHF);
340 0910 4         FIRST_CHAR = 0;
341 0911 4
342 0912 4         IF (.EDT$$A_SEL_BUF EQL .EDT$$A_CUR_BUF) !
343 0913 4         THEN
344 0914 4             EDT$$$C_REVIDCHK (CH$DIFF (.TXTPTR, CH$PTR (EDT$$A_WK_LN [LIN_TEXT])) - 1);
345 0915 4
346 0916 4         EDT$$$C_SHWBLOB ();
347 0917 4         EDT$$FMT_CH (XC' ');
348 0918 4
349 0919 4         IF (.CHAR EQL ASC_K_TAB)
350 0920 4         THEN
351 0921 4             BEGIN
352 0922 4             CHAR_WIDTH = .CHAR_WIDTH - 2;
353 0923 4
```

```
354 0924 4      ASSERT (.CHAR_WIDTH EQL 6);
355 0925 4      END;
356 0926 4
357 0927 4      END;
358 0928 4
359 0929 4      MAXPOS = .SCRPTR [SCR_EDIT_MAXPOS];
360 0930 4
361 0931 4      + This is the loop that actually puts characters into the format buffer for output to the screen.
362 0932 4      - The time around this loop is critical to EDT's performance in screen mode.
363 0933 4
364 0934 4      WHILE ((.LEN GTR 0) AND (.EDT$G_FMT_LNPOS LSS (.WIDTH - .CHAR_WIDTH)) AND !
365 0935 4      ((.TXTPTR - .ORIG_TXTPTR - 1) LEQ .MAXPOS)) DO
366 0936 4      BEGIN
367 0937 4
368 0938 4      IF (.EDT$A_SEL_BUF EQL .EDT$A_CUR_BUF) !
369 0939 4      THEN
370 0940 4      EDT$SC_REVIDCHK (CH$DIFF (.TXTPTR, CH$PTR (EDT$A_WK_LN [LIN_TEXT])) - 1);
371 0941 4
372 0942 4      IF (.EDT$G_FMT_LNPOS GEQ .EDT$G_SHF)
373 0943 4      THEN
374 0944 4      BEGIN
375 0945 4
376 0946 4      IF .FIRST_CHAR
377 0947 4      THEN
378 0948 4      BEGIN
379 0949 4      EDT$SC_POSCSIF (.EDT$G_CS_LNO, .EDT$G_FMT_LNPOS - .EDT$G_SHF);
380 0950 4      FIRST_CHAR = 0;
381 0951 4      END;
382 0952 4
383 0953 4      + Put the character in the format buffer.
384 0954 4      - Do simple characters in-line; call EDT$FMT_CH for complex characters.
385 0955 4
386 0956 4
387 0957 4
388 0958 4
389 0959 4      IF .SIMPLE_CHAR
390 0960 4      THEN
391 0961 4      BEGIN
392 0962 4      EDT$G_FMT_LNPOS = .EDT$G_FMT_LNPOS + 1;
393 0963 4
394 0964 4      IF (.EDT$A_FMT_CUR EQLA CH$PTR (EDT$T_FMT_BUF, EDT$K_FMT_BUFLN))
395 0965 4      THEN
396 0966 4      BEGIN
397 0967 4      + We have reached the end of the buffer; empty it.
398 0968 4      -
399 0969 4
400 0970 4
401 0971 4      LOCAL
402 0972 4      SAV_LNPOS;
403 0973 4
404 0974 4      SAV_LNPOS = .EDT$G_FMT_LNPOS;
405 0975 4      EDT$OUT_FMTBUF ();
406 0976 4      EDT$G_FMT_LNPOS = .SAV_LNPOS;
407 0977 4      END;
408 0978 4
409 0979 4      CH$WCHAR_A (.CHAR, EDT$A_FMT_CUR);
410 0980 4
```

```
411 0981      IF (.EDT$G_PRV_COL NEQ (.EDT$G_TI_WID - 1)) THEN EDT$G_PRV_COL = .EDT$G_PRV_COL + 1;
412 0982
413 0983      END
414 0984      ELSE
415 0985          EDT$FMT_CH (.CHAR);
416 0986
417 0987      END
418 0988      ELSE
419 0989          EDT$G_FMT_LNPOS = .EDT$G_FMT_LNPOS + .CHAR_WIDTH;
420 0990
421 0991      LEN = .LEN - 1;
422 0992      CHAR = CH$RCHAR_A (TXTPTR);
423 0993
424 0994      IF ((.CHAR GEQ 'X'20') AND (.CHAR LEQ 'X'7E'))
425 0995      THEN
426 0996          BEGIN
427 0997              CHAR_WIDTH = 1;
428 0998              SIMP[E_CHAR = 1;
429 0999          END
430 1000      ELSE
431 1001          BEGIN
432 1002              CHAR_WIDTH = EDT$FMT_CHWID (.CHAR, .EDT$G_FMT_LNPOS);
433 1003              SIMP[E_CHAR = 0;
434 1004          END;
435 1005
436 1006      END;
437 1007
438 1008      +
439 1009      If we have not finished the line, it may be because the line won't fit on the screen.
440 1010      Since the loop above stops one column short of the right edge of the screen, there
441 1011      may be just room for one more character; if so, put it out.  If not, put a blob in the
442 1012      last column.
443 1013
444 1014
445 1015      IF ((.LEN GTR 0) AND ((.TXTPTR - .ORIG_TXTPTR - 1) LEQ .MAXPOS))
446 1016      THEN
447 1017          BEGIN
448 1018
449 1019          IF ((.LEN EQL 1) AND (.EDT$G_FMT_LNPOS EQL (.WIDTH - .CHAR_WIDTH)) AND
450 1020              (.EDT$G_FMT_LNPOS GEQ .EDT$G_SHF))
451 1021          THEN
452 1022              BEGIN
453 1023
454 1024              IF (.EDT$A_SEL_BUF EQL .EDT$A_CUR_BUF)
455 1025              THEN
456 1026                  EDT$SC_REVIDCHK (CH$DIFF (.TXTPTR, CH$PTR (EDT$A_WK_LN [LIN_TEXT])) - 1);
457 1027
458 1028              IF .FIRST_CHAR
459 1029              THEN
460 1030                  BEGIN
461 1031                      EDT$SC_POSCSIF (.EDT$G_CS_LNO, .EDT$G_FMT_LNPOS - .EDT$G_SHF);
462 1032                      FIRST_CHAR = 0;
463 1033                  END;
464 1034
465 1035          EDT$FMT_CH (.CHAR);
466 1036          LEN = .LEN - 1;
467 1037      END
```



```
468 1038 3 ELSE
469 1039 4 BEGIN
470 1040 5
471 1041 5 IF (( NOT .ERASED) AND (.SCRPTR [SCR_EDIT_MAXPOS] EQL 255))
472 1042 5 THEN
473 1043 5 BEGIN
474 1044 5 EDT$SC_POSCSIF (.EDT$G_CS_LNO, MAX (0, .EDT$G_FMT_LNPOS - .EDT$G_SHF));
475 1045 5 EDT$SC_ERATOEOOL ();
476 1046 5 END;
477 1047 5
478 1048 5 +
479 1049 5 | If there is room left on the line, it may be that we have printed no characters.
480 1050 5 | Therefore, print a space to be sure that the VT100's autowrap flag is not set.
481 1051 5 |
482 1052 5
483 1053 5 IF (.EDT$G_FMT_LNPOS LSS (.EDT$G_TI_WID - 1)) THEN EDT$FMT_CH (XC' ');
484 1054 5
485 1055 5 EDT$SC_POSCSIF (.EDT$G_CS_LNO, .EDT$G_TI_WID - 1);
486 1056 5 EDT$SC_SHWBLOB ();
487 1057 5 END;
488 1058 5
489 1059 5 END
490 1060 5 +
491 1061 5 | Throw in an erase to end of line sequence if we have painted as close as we can to the right margin.
492 1062 5 | Suppress the sequence if we have just put a character at the right margin or if the line is already erased
493 1063 5 |
494 1064 5 ELSE
495 1065 5
496 1066 5 IF (( NOT .ERASED) AND (.SCRPTR [SCR_EDIT_MAXPOS] EQL 255))
497 1067 5 THEN
498 1068 5 BEGIN
499 1069 5
500 1070 5 IF .FIRST_CHAR THEN EDT$SC_POSCSIF (.EDT$G_CS_LNO, MAX (0, .EDT$G_FMT_LNPOS - .EDT$G_SHF));
501 1071 5
502 1072 5 EDT$SC_ERATOEOOL ();
503 1073 5 END;
504 1074 5
505 1075 5 +
506 1076 5 | Mark the line as finished with its edit.
507 1077 5 |
508 1078 5 SCRPTR [SCR_EDIT_MINPOS] = MIN (.SCRPTR [SCR_CHR_TO] - .SCRPTR [SCR_CHR_FROM] + 1, 255);
509 1079 5 SCRPTR [SCR_EDIT_MAXPOS] = 0;
510 1080 5 SCRPTR [SCR_EDIT_FLAGS] = .SCRPTR [SCR_EDIT_FLAGS] AND ( NOT (SCR_EDIT_MODIFY OR SCR_EDIT_INSLN));
511 1081 5 END;
! of routine EDT$SC_RFRELN
```

.TITLE EDT\$SCRRLIN EDT\$SCRRLIN - refresh a screen line  
.IDENT \V04-000\

.EXTRN EDT\$FMT\_CH, EDT\$FMT\_CHWID  
.EXTRN EDT\$SC\_SHWBLOB  
.EXTRN EDT\$SC\_REVIDCHK  
.EXTRN EDT\$SC\_NONREVID  
.EXTRN EDT\$SC\_POSCSIF  
.EXTRN EDT\$SC\_ERATOEOOL  
.EXTRN EDT\$SC\_ERAALL, EDT\$FMT\_TEXT  
.EXTRN EDT\$OUT\_FMTBUF

				OFFC 00000		
	SE		04	C2	00002	
		00000000G	00	D5	00005	
			07	13	0000B	
00000000G	00		00	FB	0000D	
	54	04	AC	D0	00014	1\$:
00000000G	00		54	D1	00018	
			38	12	0001F	
			7E	D4	00021	
		00000000G	00	DD	00023	
00000000G	00		02	FB	00029	
00000000G	00		00	FB	00030	
			7E	D4	00037	
00000000G	00		01	FB	00039	
	0E	08	AC	E8	00040	
FF	BF	0C	A4	91	00044	
			07	12	00049	
00000000G	00		00	FB	0004B	
0B	A4		01	8E	00052	2\$:
			03	6F	00056	
53	00000000G	00	00	C1	00059	3\$:
		09	A4	9A	00065	
		0A	A4	9A	00069	
			52	D6	0006D	
52	60	50	00	D0	0006F	
		08	00	ED	00076	
			03	18	0007B	
		52	60	9A	0007D	
	5B	52	51	C3	00080	4\$:
		52	9E	00084		
		6E	D0	00089		
			00	D4	0008C	
		00000000G	69	11	00092	
			00	DD	00094	5\$:
			59	DD	0009A	
00000000G	00		02	FB	0009C	
	55		50	D0	000A3	
			57	D4	000A6	
50	0B	50	6E	C3	000A8	6\$:
	A4	52	00	ED	000AC	
		08	62	19	000B2	
			5B	D5	000B4	
			5E	15	000B6	
	50	53	55	C3	000B8	

  

		EDT\$\$\$SC REP MODE	
.EXTRN	EDT\$\$\$A_EOB_SCRPTR		
.EXTRN	EDT\$\$\$G_CS_LNO, EDT\$\$\$A_SEL_BUF		
.EXTRN	EDT\$\$\$G_SHF, EDT\$\$\$G_TI_WID		
.EXTRN	EDT\$\$\$A_WK_LN, EDT\$\$\$G_FMT_LNPOS		
.EXTRN	EDT\$\$\$A_CUR_BUF, EDT\$\$\$A_FMT_CUR		
.EXTRN	EDT\$\$\$T_FMT_BUF, EDT\$\$\$G_PRV_COL		
.EXTRN	EDT\$\$\$G_INSERT_MODE		
.EXTRN	EDT\$\$\$INTER_ERR		
.PSECT	_EDT\$CODE, NOWRT, SHR, PIC, 2		
.ENTRY	EDT\$\$\$SC_RFRELN, Save R2,R3,R4,R5,R6,R7,R8,-	0698	
	R9,R10,R11		
SUBL2	#4, SP		
TSTL	EDT\$\$\$G_INSERT_MODE	0794	
BEQL	1\$		
CALLS	#0, EDT\$\$\$SC_REP_MODE		
MOVL	SCRPTR, R4	0800	
CMPL	R4, EDT\$\$\$A_EOB_SCRPTR		
BNEQ	3\$		
CLRL	-(SP)	0803	
PUSHL	EDT\$\$\$G_CS_LNO		
CALLS	#2, EDT\$\$\$SC_POSCSIF		
CALLS	#0, EDT\$\$\$SC_NONREVID	0804	
CLRL	-(SP)	0805	
CALLS	#1, EDT\$\$\$FMT_TEXT		
BLBS	ERASED, 2\$	0808	
CMPB	12(R4), #255		
BNEQ	2\$		
CALLS	#0, EDT\$\$\$SC_ERATOEOI	0812	
MNEGB	#1, 11(R4)	0819	
BRW	38\$	0820	
ADDL3	EDT\$\$\$G_SHF, EDT\$\$\$G_TI_WID, WIDTH	0829	
MOVZBL	9(R4), LEFT	0830	
MOVZBL	10(R4), R2	0831	
INCL	R2		
MOVL	EDT\$\$\$A_WK_LN, R0		
CMPZV	#0, #8, (R0), R2		
BGEQ	4\$		
MOVZBL	(R0), R2		
SUBL3	LEFT, R2, LEN		
MOVAB	7(LEFT)(R0), TXTPTR	0832	
MOVL	TXTPTR, ORIG_TXTPTR	0833	
CLRL	EDT\$\$\$G_FMT_LNPOS	0834	
BRB	8\$	0835	
PUSHL	EDT\$\$\$G_FMT_LNPOS	0845	
PUSHL	CHAR		
CALLS	#2, EDT\$\$\$FMT_CHWID		
MOVL	R0, CHAR_WIDTH		
CLRL	SIMPLE_CHAR	0846	
SUBL3	ORIG_TXTPTR, TXTPTR, R0	0853	
CMPZV	#0, #8, 11(R4), R0		
BLSS	9\$		
TSTL	LEN	0854	
BLEQ	9\$		
SUBL3	CHAR_WIDTH, WIDTH, R0	0855	

		50	D7	000BC	DECL	R0	
	50	00000000G	00	D1	000BE	CMPL	EDT\$G_FMT_LNPOS, R0
			4F	18	000C5	BGEQ	9\$
		00000000G	00	D5	000C7	TSTL	EDT\$G_FMT_LNPOS
			25	12	000CD	BNEQ	7\$
		08	A4	95	000CF	TSTB	8(R4)
			20	13	000D2	BEQL	7\$
00000000G	00	00000000G	00	C1	000D4	ADDL3	#2, EDT\$G_SHF, EDT\$G_FMT_LNPOS
	09		59	D1	000E0	CMPL	CHAR, #9
			0F	12	000E3	BNEQ	7\$
	55		02	C2	000E5	SUBL2	#2, CHAR WIDTH
	06		55	D1	000E8	CMPL	CHAR_WIDTH, #6
			07	13	000EB	BEQL	7\$
	00000000G	00	00	FB	000ED	CALLS	#0, EDT\$INTER_ERR
	00000000G	00	55	C0	000F4	ADDL2	CHAR_WIDTH, EDT\$G_FMT_LNPOS
			58	D7	000FB	DECL	LEN
	59		82	9A	000FD	MOVZBL	(TXTPTR)+, CHAR
	20		59	D1	00100	CMPL	CHAR, #32
			8F	19	00103	BLSS	5\$
0000007E	8F		59	D1	00105	CMPL	CHAR, #126
			86	14	0010C	BGTR	5\$
	55		01	D0	0010E	MOVL	#1, CHAR WIDTH
	57		01	D0	00111	MOVL	#1, SIMPLE_CHAR
			92	11	00114	BRB	6\$
	58		01	D0	00116	MOVL	#1, FIRST_CHAR
		08	A4	95	00119	TSTB	8(R4)
			70	13	0011C	BEQL	11\$
		00000000G	00	D5	0011E	TSTL	EDT\$G_FMT_LNPOS
			68	12	00124	BNEQ	11\$
	50	00000000G	00	D0	00126	MOVL	EDT\$G_SHF, R0
			50	D0	0012D	MOVL	R0, EDT\$G_FMT_LNPOS
7E	00000000G	00	50	C3	00134	SUBL3	R0, EDT\$G_FMT_LNPOS, -(SP)
		00000000G	00	DD	0013C	PUSHL	EDT\$G_CS [NO
	00000000G	00	02	FB	00142	CALLS	#2, EDT\$SC_POSCSIF
			58	D4	00149	CLRL	FIRST_CHAR
	00000000G	00	00	D1	0014B	CMPL	EDT\$A_SEL_BUF, EDT\$A_CUR_BUF
			12	12	00156	BNEQ	10\$
50	52	00000000G	00	C3	00158	SUBL3	EDT\$A_WK_LN, TXTPTR, R0
		F8	A0	9F	00160	PUSHAB	-8(R0)
	00000000G	00	01	FB	00163	CALLS	#1, EDT\$SC_REVIDCHK
	00000000G	00	00	FB	0016A	CALLS	#0, EDT\$SC_SHWBLOB
			20	DD	00171	PUSHL	#32
	00000000G	00	01	FB	00173	CALLS	#1, EDT\$FMT_CH
		09	59	D1	0017A	CMPL	CHAR, #9
			0F	12	0017D	BNEQ	11\$
	55		02	C2	0017F	SUBL2	#2, CHAR WIDTH
	06		55	D1	00182	CMPL	CHAR_WIDTH, #6
			07	13	00185	BEQL	11\$
	00000000G	00	00	FB	00187	CALLS	#0, EDT\$INTER_ERR
		5A	A4	9A	0018E	MOVZBL	12(R4), MAXPOS
			58	D5	00192	TSTL	LEN
			03	14	00194	BGTR	14\$
		00FB	31	00196	BRW	23\$	
50	53		55	C3	00199	SUBL3	CHAR WIDTH, WIDTH, R0
	50	00000000G	00	D1	0019D	CMPL	EDT\$G_FMT_LNPOS, R0
			F0	18	001A4	BGEQ	13\$
50	52		6E	C3	001A6	SUBL3	ORIG_TXTPTR, TXTPTR, R0



		50	D7	001AA	DECL	RO		
	SA	50	D1	001AC	CMPL	RO	MAXPOS	
		E5	14	001AF	BGTR	13\$		
	00000000G	00	D1	001B1	CMPL	EDT\$SA_SEL_BUF, EDT\$SA_CUR_BUF		0939
50		12	12	001BC	BNEQ	15\$		
	52	00	C3	001BE	SUBL3	EDT\$SA_WK_LN, TXTPTR, RO		0941
		AO	9F	001C6	PUSHAB	-8(RO)		
	00000000G	00	FB	001C9	CALLS	#1, EDT\$SSC REVIDCHK		
		51	00	001D0	MOVL	EDT\$SG_FMT_LNPOS, R1		0943
		50	00	001D7	MOVL	EDT\$SG_SHF, RO		
		50	D1	001DE	CMPL	R1, RO		
		78	19	001E1	BLSS	19\$		
	13		58	E9	001E3	BLBC	FIRST_CHAR, 16\$	0947
7E		51	50	C3	001E6	SUBL3	RO, RT, -(SP)	0950
	00000000G	00	DD	001EA	PUSHL	EDT\$SG_CS_LNO		
		02	FB	001F0	CALLS	#2, EDT\$SSC_POSCSIF		
		58	D4	001F7	CLRL	FIRST_CHAR		0951
	54		57	E9	001F9	BLBC	SIMPLE_CHAR, 18\$	0959
		00	D6	001FC	INCL	EDT\$SG_FMT_LNPOS		0962
	00000000G	00	9E	00202	MOVAB	EDT\$ST_FMT_BUF+512, RO		0964
		50	00	D1	00209	CMPL	EDT\$SA_FMT_CUR, RO	
		15	12	00210	BNEQ	17\$		
	56	00000000G	00	D0	00212	MOVL	EDT\$SG_FMT_LNPOS, SAV_LNPOS	0974
		00	00	FB	00219	CALLS	#0, EDT\$SODT_FMTBUF	0975
	00000000G	00	56	D0	00220	MOVL	SAV_LNPOS, EDT\$SG_FMT_LNPOS	0976
		50	00	D0	00227	MOVL	EDT\$SA_FMT_CUR, RO	0979
		60	59	90	0022E	MOVB	CHAR, (RO)	
	00000000G	00	D6	00231	INCL	EDT\$SA_FMT_CUR		
50	00000000G	00	01	C3	00237	SUBL3	#1, EDT\$SG_TI_WID, RO	0981
		50	00	D1	0023F	CMPL	EDT\$SG_PRV_COL, RO	
		1A	13	00246	BEQL	20\$		
	00000000G	00	D6	00248	INCL	EDT\$SG_PRV_COL		
		12	11	0024E	BRB	20\$		0959
		59	DD	00250	PUSHL	CHAR		0985
	00000000G	00	01	FB	00252	CALLS	#1, EDT\$SFMT_CH	
		07	11	00259	BRB	20\$		0943
	00000000G	00	55	C0	0025B	ADDL2	CHAR_WIDTH, EDT\$SG_FMT_LNPOS	0989
		58	D7	00262	DECL	LEN		0991
		59	82	9A	00264	MOVZBL	(TXTPTR)+, CHAR	0992
		20	59	D1	00267	CMPL	CHAR, #32	0994
			11	19	0026A	BLSS	21\$	
	0000007E	8F	59	D1	0026C	CMPL	CHAR, #126	
			08	14	00273	BGTR	21\$	
		55	01	D0	00275	MOVL	#1, CHAR_WIDTH	0997
		57	01	D0	00278	MOVL	#1, SIMPLE_CHAR	0998
			14	11	0027B	BRB	22\$	0994
	00000000G	00	DD	0027D	PUSHL	EDT\$SG_FMT_LNPOS		1002
		59	DD	00283	PUSHL	CHAR		
	00000000G	00	02	FB	00285	CALLS	#2, EDT\$SFMT_CHWID	
		55	50	D0	0028C	MOVL	RO, CHAR_WIDTH	
			57	D4	0028F	CLRL	SIMPLE_CHAR	1003
		FEFE	31	00291	BRW	12\$		0935
			58	D5	00294	TSTL	LEN	1015
		03	14	00296	BGTR	25\$		
		00DB	31	00298	BRW	33\$		
50			6E	C3	0029B	SUBL3	ORIG_TXTPTR, TXTPTR, RO	
	52		50	D7	0029F	DECL	RO	

5A	50	D1	002A1	CMPL	R0, MAXPOS	
01	F2	14	002A4	BGTR	24\$	1019
53	5B	D1	002A6	CMPL	LEN, #1	
53	62	12	002A9	BNEQ	28\$	
53	55	C2	002AB	SUBL2	CHAR WIDTH, R3	
00000000G	00	D1	002AE	CMPL	EDTSSG_FMT_LNPOS, R3	
00000000G	56	12	002B5	BNEQ	28\$	
00000000G	00	D1	002B7	CMPL	EDTSSG_FMT_LNPOS, EDTSSG_SHF	1020
00000000G	49	19	002C2	BLSS	28\$	
00000000G	00	D1	002C4	CMPL	EDTSSA_SEL_BUF, EDTSSA_CUR_BUF	1024
52	11	12	002CF	BNEQ	26\$	
00000000G	00	C2	002D1	SUBL2	EDTSSA_WK_LN, R2	1026
00000000G	00	A2	9F 002D8	PUSHAB	-8(R2)	
00000000G	00	01	FB 002D8	CALLS	#1, EDTSSSC_REVIDCHK	
7E	1B	58	E9 002E2	BLBC	FIRST CHAR, 27\$	1028
00000000G	00	00	C3 002E5	SUBL3	EDTSSG_SHF, EDTSSG_FMT_LNPOS, -(SP)	1031
00000000G	00	00	DD 002F1	PUSHL	EDTSSG_CS_LNO	
00000000G	00	02	FB 002F7	CALLS	#2, EDTSSSC_POSCSIF	
00000000G	00	58	D4 002FE	CLRL	FIRST_CHAR	1032
00000000G	00	59	DD 00300	PUSHL	CHAR	1035
00000000G	00	01	FB 00302	CALLS	#1, EDTSSFMT_CH	
00000000G	00	5B	D7 00309	DECL	LEN	1036
00000000G	00	67	11 0030B	BRB	32\$	1019
00000000G	00	AC	E8 0030D	BLBS	ERASED, 30\$	1041
00000000G	00	A4	91 00311	CMPB	12(R4), #255	
00000000G	00	26	12 00316	BNEQ	30\$	
50	00000000G	00	C3 00318	SUBL3	EDTSSG_SHF, EDTSSG_FMT_LNPOS, R0	1044
00000000G	00	50	DD 00324	PUSHL	R0	
00000000G	00	02	18 00326	BGEQ	29\$	
00000000G	00	6E	D4 00328	CLRL	(SP)	
00000000G	00	00	DD 0032A	PUSHL	EDTSSG_CS_LNO	
00000000G	00	02	FB 00330	CALLS	#2, EDTSSSC_POSCSIF	
00000000G	00	00	FB 00337	CALLS	#0, EDTSSSC_ERATOEOI	1045
50	00000000G	00	01 C3 0033E	SUBL3	#1, EDTSSG_TI_WID, R0	1053
00000000G	50	00	D1 00346	CMPL	EDTSSG_FMT_LNPOS, R0	
00000000G	00	09	18 0034D	BGEQ	31\$	
00000000G	00	20	DD 0034F	PUSHL	#32	
7E	00000000G	00	01 FB 00351	CALLS	#1, EDTSSFMT_CH	
00000000G	00	01	C3 00358	SUBL3	#1, EDTSSG_TI_WID, -(SP)	1055
00000000G	00	00	DD 00360	PUSHL	EDTSSG_CS_LNO	
00000000G	00	02	FB 00366	CALLS	#2, EDTSSSC_POSCSIF	
00000000G	00	00	FB 0036D	CALLS	#0, EDTSSSC_SHWBLOB	1056
00000000G	00	34	11 00374	BRB	36\$	1015
00000000G	00	AC	E8 00376	BLBS	ERASED, 36\$	1066
00000000G	00	A4	91 0037A	CMPB	12(R4), #255	
00000000G	00	29	12 0037F	BNEQ	36\$	
00000000G	00	58	E9 00381	BLBC	FIRST CHAR, 35\$	1070
50	00000000G	00	C3 00384	SUBL3	EDTSSG_SHF, EDTSSG_FMT_LNPOS, R0	
00000000G	00	50	DD 00390	PUSHL	R0	
00000000G	00	02	18 00392	BGEQ	34\$	
00000000G	00	6E	D4 00394	CLRL	(SP)	
00000000G	00	00	DD 00396	PUSHL	EDTSSG_CS_LNO	
00000000G	00	02	FB 0039C	CALLS	#2, EDTSSSC_POSCSIF	
00000000G	00	00	FB 003A3	CALLS	#0, EDTSSSC_ERATOEOI	1072
00000000G	00	A4	9A 003AA	MOVZBL	10(R4), R0	1078
00000000G	00	A4	9A 003AE	MOVZBL	9(R4), R1	
00000000G	00	51	C2 003B2	SUBL2	R1, R0	

EDT\$SCRRLIN  
V04-000

EDT\$SCRRLIN - refresh a screen line  
EDT\$SC\_RFRELN - refresh a line on the screen

E 11  
16-Sep-1984 01:42:29  
14-Sep-1984 12:24:38

VAX-11 Bliss-32 V4.0-742  
DISK\$VMSMASTER:[EDT.SRC]SCRRLIN.BLI;1

Page 15  
(3)

000000FF	8F	50	D6	003B5	INCL	R0	
		50	D1	003B7	CMPL	R0	#255
		04	15	003BE	BLEQ	374	
	50	FF	8F	9A	003C0	MOVZBL	#255, R0
0B	A4		50	90	003C4	MOVB	R0, 11(R4)
0C	A4	03FF	8F	AA	003C8	BICW2	#1023, 12(R4)
			04	003CE	RET		

1080  
1081

; Routine Size: 975 bytes, Routine Base: \_EDT\$CODE + 0000

; 512 1082 1  
; 513 1083 1 !<BLF/PAGE>



EDT\$SCRRLIN  
V04-000

EDT\$SCRRLIN - refresh a screen line  
EDT\$SSC\_RFRELN - refresh a line on the screen

F 11  
16-Sep-1984 01:42:29  
14-Sep-1984 12:24:38

VAX-11 Bliss-32 V4.0-742  
DISK\$VMSMASTER:[EDT.SRC]SCRRLIN.BLI;1

Page 16  
(4)

: 515 1084 1 END  
: 516 1085 1  
: 517 1086 0 ELUDOM

! of module EDT\$SCRRLIN

### PSECT SUMMARY

Name	Bytes	Attributes
_EDT\$CODE	975	NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

### Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
\$255\$DUA28:[EDT.SRC]EDT.L32;1	377	48	12	40	00:00.2
_ \$255\$DUA28:[EDT.SRC]PSECTS.L32;1	2	1	50	7	00:00.1

### COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACEBACK/LIS=LIS\$:SCRRLIN/OBJ=OBJ\$:SCRRLIN MSRC\$:SCRRLIN.BLI/UPDATE=(ENH\$:SCRRLIN)

Size: 975 code + 0 data bytes  
Run Time: 00:36.5  
Elapsed Time: 00:43.0  
Lines/CPU Min: 1787  
Lexemes/CPU-Min: 7088  
Memory Used: 246 pages  
Compilation Complete



0139 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

